

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ  
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ  
ТЕХНОЛОГІЙ**

**ФОРМА НАВЧАННЯ ЗАОЧНА  
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ  
ІНФОРМАТИКИ**

**Допускається до захисту**

Завідувач кафедри \_\_\_\_\_ О.О. Ємець  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2021 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО ДИПЛОМНОЇ РОБОТИ**

**на тему**

**СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ З ТЕМИ  
«МАТЕМАТИЧНІ ОСНОВИ ДИСЦИПЛІНИ» ДИСТАНЦІЙНОГО  
НАВЧАЛЬНОГО КУРСУ «ТЕОРІЯ ПРОГРАМУВАННЯ»**

**зі спеціальності 122 «Комп'ютерні науки»**

**Виконавець роботи** Коломієць Євгеній Сергійович

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 2021р.  
(підпис)

**Науковий керівник** к.ф.-м.н., доц. Черненко Оксана Олексіївна

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 2021р.  
(підпис)

**ПОЛТАВА 2021 р.**

## ЗМІСТ

ВСТУП .....	3
1. ПОСТАНОВКА ЗАДАЧІ.....	5
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	7
2.1. Огляд програмного забезпечення для дистанційного навчання .....	7
2.2. Актуальність використання дистанційного навчання.....	14
3. ТЕОРЕТИЧНА ЧАСТИНА .....	18
3.1. Огляд матеріалу за темою роботи .....	18
3.2. Алгоритмізація задачі за темою роботи .....	30
3.3. Обґрунтування вибору програмних засобів для реалізації завдання роботи.....	43
4. ПРАКТИЧНА ЧАСТИНА .....	45
4.1. Розробка блок-схеми, яка підлягає програмуванню .....	45
4.2. Опис процесу програмної реалізації .....	47
4.3. Опис роботи тренажеру.....	51
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	56
ДОДАТОК А.....	58

## ВСТУП

Дистанційне навчання передбачає високотехнологічний підхід до процесу передачі знань і дає можливість створення системи масового безперервного самонавчання, загального обміну інформацією. Саме ця система може найбільш адекватно і гнучко реагувати на потреби суспільства щодо підготовки високопрофесійних фахівців. Крім того, впровадження дистанційного навчання уможлиблює на практиці ідею пожиттєвої освіти, навчання через усе життя.

Система дистанційної освіти підвищує ефективність і якість навчання завдяки додатковим можливостям пізнання навколишньої дійсності і самопізнання, розвитку особистості студента; управління навчально-виховним процесом, проведенню моніторингу (контролю, корекції результатів навчальної діяльності, комп'ютерного педагогічного тестування і психодіагностики; поширення науково-методичного досвіду; організації інтелектуального дозвілля.

Модель інноваційного університету нині зазнає досить потужного впливу мережевої освіти. Дистанційне навчання варто сприймати не як реального суперника традиційної університетської освіти, а як одне з новітніх досягнень інформаційних технологій, яке потенційно може змінити характер усталених віками освітніх практик, розширивши та доповнивши можливості класичної системи навчання.

Метою роботи є створення програмного забезпечення тренажеру з теми «Математичні основи дисципліни» дистанційного навчального курсу «Теорія програмування».

Об'єктом розробки є процес дистанційного навчання математичним дисциплінам.

Предметом розробки є програма тренажеру з теми «Математичні основи дисципліни».

Перелік використаних методів:

- застосування математичних основ теорії програмування;
- безсерверну платформу Netlify;
- фреймворк Vue.js;
- бібліотека vue-router;
- UI-фреймворк Quasar.

Тренажер готовий до використання в дистанційному курсі «Теорія програмування».

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі для реалізації тренажеру. У другому розділі описано огляд програмного забезпечення для дистанційного навчання, вказано актуальність використання дистанційного навчання. В третьому представлено огляд матеріалу за темою роботи, алгоритмізацію задачі за темою роботи, обґрунтування вибору програмних засобів для реалізації завдання роботи. В четвертому – розробка блок-схеми, яка підлягає програмуванню, опис процесу програмної реалізації, опис роботи тренажеру.

Обсяг пояснювальної записки: 58 стор., в т.ч. основна частина - 50 стор., джерела - 12 назв.

## 1. ПОСТАНОВКА ЗАДАЧІ

Планується розробка тренажеру за алгоритмом, який розробляється в рамках роботи, та його подальше використання як складової дистанційного курсу «Теорія програмування». Отже, необхідно врахувати можливість інтеграцій тренажеру до системи дистанційного навчання Moodle.

Розглянемо основні завдання роботи:

- провести аналіз переваг використання тренажерів
- провести вибір програмних засобів для реалізації програми;
- розглянути теоретичні відомості за обраною темою;
- розробити алгоритм тренажеру з теми «Математичні основи дисципліни»;
- розробити тренажер;
- описати процес програмної реалізації тренажеру;
- описати роботу програми на прикладі.

Слід розробити можливість спочатку переглянути теоретичний матеріал за темою, та за необхідності його завантажити. На кожному кроці тренажеру повинно виводитися завдання, відповідь має надаватися одним із способів:

- вибір одного з декількох варіантів;
- вибір декількох варіантів;
- встановлення відповідності.

При неправильній відповіді слід відображати повідомлення про помилку.

Для використання тренажера іноземними студентами в рамках вивчення дистанційного курсу «Теорія програмування» потрібно розробити можливість зміни мови у програмі або запуску тренажеру відразу на англійській мові.



## 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

### 2.1. Огляд програмного забезпечення для дистанційного навчання

Важливою складовою частиною дистанційного навчання є його реалізація за допомогою використання інформаційних технологій, а саме системи управління навчанням, створеної для розробки, управління та поширення навчальних матеріалів онлайн із забезпеченням спільного доступу багатьох користувачів. В англomовній літературі можна зустріти таку аббревіатуру систем управління навчанням:

- LMS – Learning Management System (система управління навчанням);
- CMS – Course Management System (система управління курсами);
- LCMS – Learning Content Management System (система управління навчальним матеріалом);
- MLE – Managed Learning Environment (оболонка для управління навчанням);
- LSS – Learning Support System (система підтримки навчання);
- LP – Learning Platform (освітня платформа);
- VLE – Virtual Learning Environments (віртуальні навчальні середовища).

Найбільш поширеними системами управління є LMS та CMS.

Сьогодні створена велика кількість різноманітних систем дистанційного навчання як з відкритим кодом (умовно безкоштовних), так і платних, широкоживаних та вузькоорієнтованих. Розглянемо декілька популярних у світі систем дистанційного навчання.

*ATutor* – модульна система дистанційним керуванням навчанням з відкритим кодом [2] створена канадськими розробниками. Поширюється

на основі GNU General Public License. Система розроблена із врахуванням доступності та можливістю адаптації за бажанням користувача.

Програмний продукт є простим у встановленні, налаштуванні та підтримці для системних адміністраторів. Викладачі можуть досить легко створювати та переносити навчальні матеріали та запускати свої онлайн-курси.

Оскільки система є модульна, тобто складається з окремих функціональних одиниць — модулів, то вона відкрита для модернізації і розширення функціональних можливостей.

Щодо операційної системи сервера, обмежень немає — система є кросплатформеною.

До недоліків системи можна віднести слабо розвинену систему звітності.

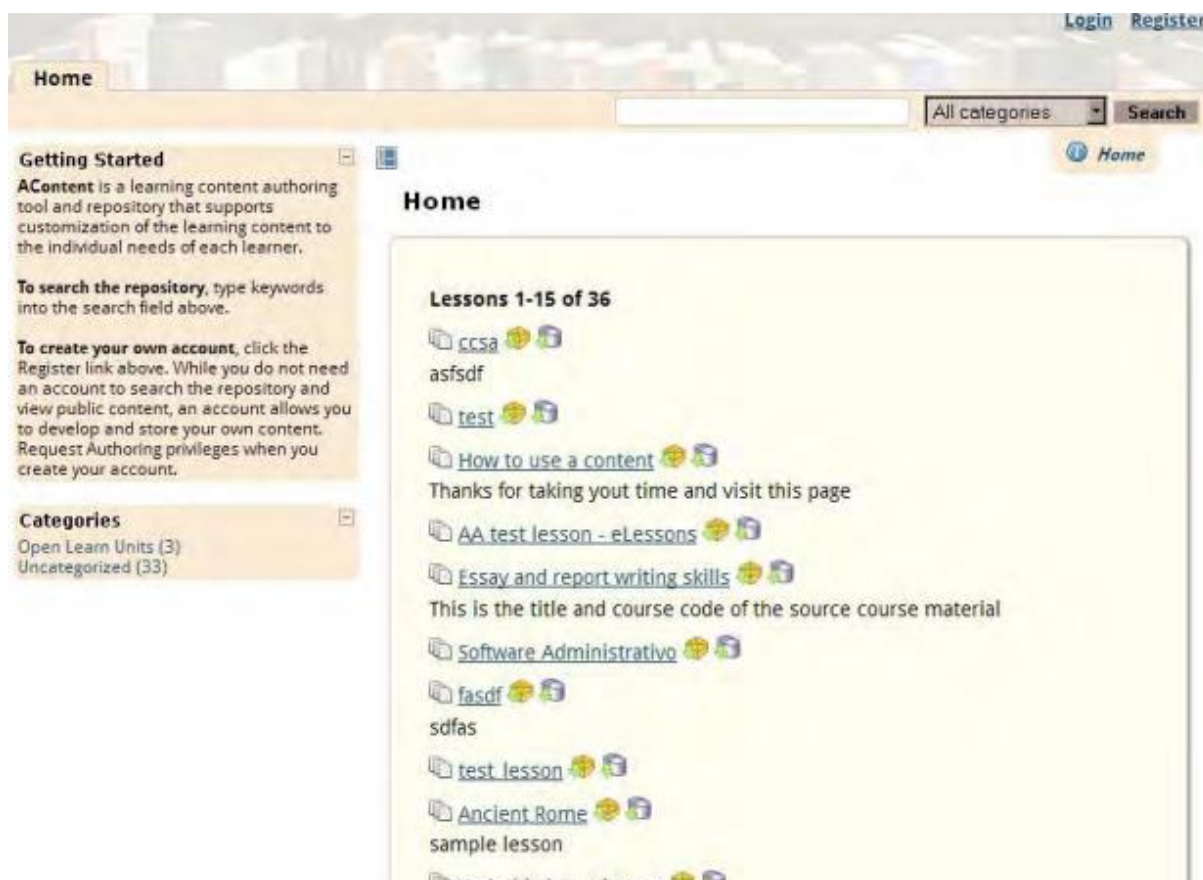


Рис.2.1 – Демонстраційний сайт системи ATutor



*Lotus Learning Space* розроблена компанією IBM, надає можливість учитися й викладати в асинхронному режимі, звертаючись до матеріалів курсів у зручний час, брати участь в онлайн-заняттях у режимі реального часу. Викладач може створювати зміст курсу в будь-яких програмах і потім розміщувати створений матеріал у Learning Space [3]. Програма має гнучку систему редагування й адміністрування курсу, дозволяє обирати різні режими викладання й відстежувати поточні результати студентів. Курси організовані у вигляді послідовних занять, які можуть бути самостійними, інтерактивними або колективними. Самостійні заняття зазвичай містять матеріали для читання і тести, які необхідно виконати після вивчення матеріалу. Інтерактивні заняття включають лекції у віртуальному класі, участь в онлайнівій дискусії або чаті, роботу з віртуальною дошкою (Whiteboard) і системою сумісного перегляду Web-сайтів (Follow me). Інтерактивні заняття плануються на певну дату і часто проводяться викладачем у віртуальному класі в режимі реального часу. Поточні результати студентів (етап проходження курсу, оцінки, витрачений час, кількість звернень та ін.) зберігаються в базі даних. Ця інформація доступна викладачеві у будь-який час у вигляді звітів різної форми. Колективні заняття передбачають заняття в офлайнівій та онлайнівій дискусіях, чаті.

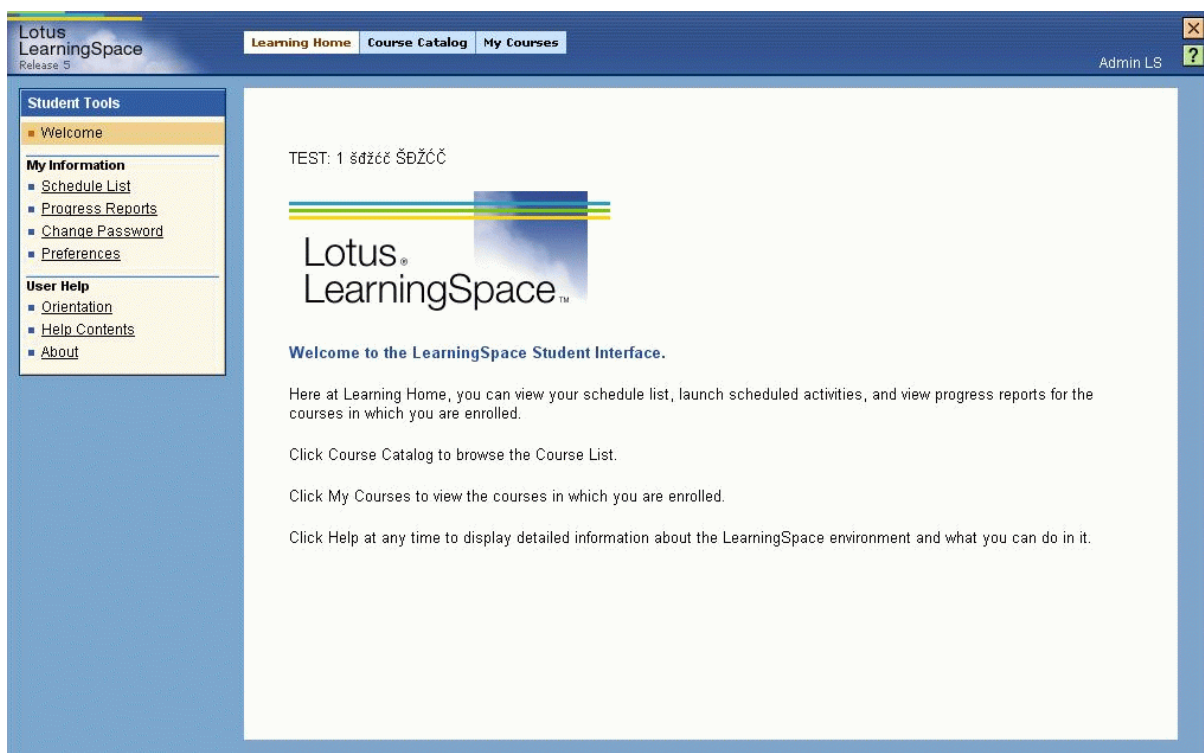


Рис.2.2 – Приклад навчального курсу в системі Lotus Learning Space

*Claroline* – платформа дистанційного навчання та електронної діяльності з відкритим кодом. Аналогічно з ATutor, поширюється на основі GNU General Public License. Сумісна з такими операційними системами, як Linux, Mac і Windows [4].

Система створена в Бельгії в Інституті педагогіки та мультимедіа Католицького університету в Луврі. Саме тому система має дуже лаконічний, достатньо зручний та зрозумілий інтерфейс. Розробленням Claroline спочатку займалися працівники навчальних закладів, її відразу створювали з урахуванням педагогічного досвіду та потреб викладачів. Інтерфейс адміністратора інтуїтивно зрозумілий, а управління та налаштування системи не потребує особливих технічних знань та навиків.

Офіційна документація, керівництво користувача для студента та викладача перекладені на російську мову.

Забезпечує інтуїтивно простий інтерфейс для адміністрування. У основу організації Claroline LMS покладено концепцію просторів,

пов'язаних з курсом чи педагогічною діяльністю. Кожен з просторів забезпечений інструментарієм для створення, організації та управління навчальними матеріалами; можливостями для забезпечення взаємодії між користувачами тощо [4].



Рис.2.3 – Приклад навчального курсу в системі Claroline

*Moodle* (Modular Object-Oriented Dynamic Learning Environment) – пакет модульного програмного забезпечення з відкритим кодом (ліцензія GNU GPL), який призначений для створення курсів дистанційного навчання та web-сайтів. Ця програма управління дистанційним навчанням, орієнтована на взаємодію між викладачем і студентом, також використовується для підтримки очних курсів. Moodle є кросплатформеним.

До основи проекту покладено п'ять принципів, які об'єднані спільною назвою «соціальний конструктивізм»:

- у сучасному навчальному середовищі ми всі одночасно є потенційними вчителями та учнями;
- ми успішні у навчанні, особливо тоді, коли намагаємось створити щось, чи пояснювати щось людям;
- розуміння інших дає змогу вивчити їх більш індивідуально;
- великий внесок у навчання робить спостереження за роботою більш досвідчених педагогів;
- навчальне середовище має бути гнучким, забезпечувати учасникам навчального процесу простий інструмент для реалізації їхніх навчальних потреб [5].

При проектуванні Moodle особлива увага приділялася таким принципам:

- просування педагогіки соціального конструкціонізму (співпраця, активне навчання, критична рефлексія тощо);
- підтримка різних підходів до навчання: дистанційне, змішане, очне;
- простий, інтуїтивно зрозумілий, ефективний, кросплатформенний інтерфейс у вікні браузера;
- проста установка на більшість платформ, що підтримують PHP;
- сумісність із більшістю широко використовуваних баз даних;
- список курсів містить описи і є доступним будь-якому користувачеві;
- курси структуруються за категоріями; – істотна увага приділяється питанням безпеки;
- для більшості текстових областей (ресурси, повідомлення форумів тощо) використовується вбудований HTML-редактор.

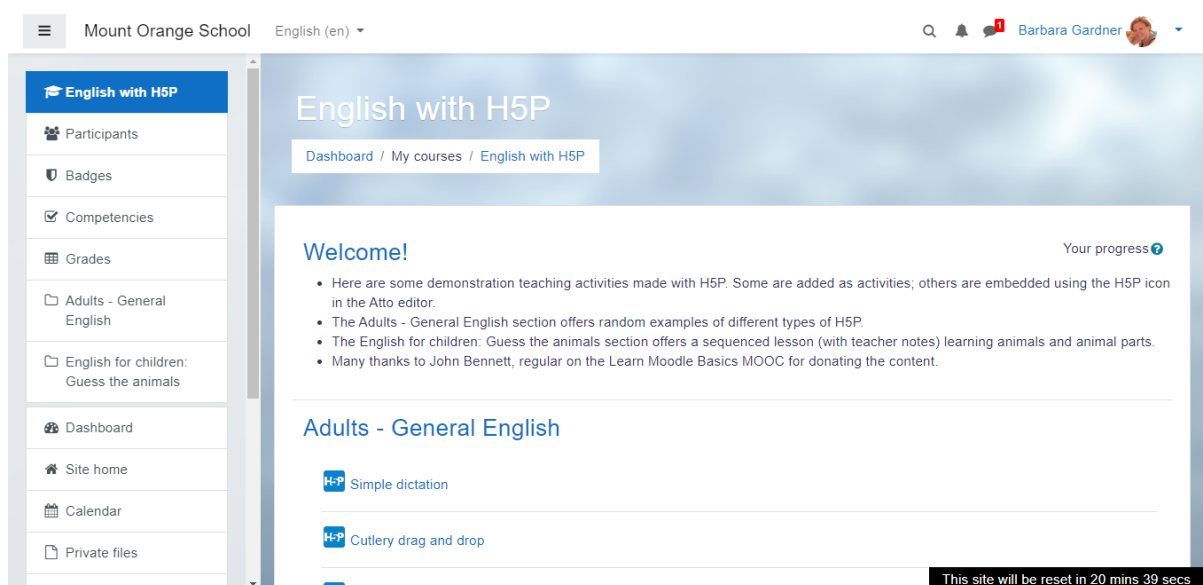


Рис.2.4 – Приклад навчальної системи в системі Moodle

Наведемо основні відомості про описані системи управління у таблиці.

Таблиця 2.1 Порівняльна характеристика деяких систем дистанційного навчання.

	Виробник	Умови поширення	Стандарти зація	Підтримка українсько ї мови	Система перевірки знань
ATutor	Канада	Вільно	IMS, SCORM	Так	Тести, контроль активності
Lotus Learning Space	США	Платно	AICC, SCORM	Так	Тести, інтерактивні заняття, контроль активності
Claroline	Бельгія	Вільно		Так	Тести, інтерактивні заняття, контроль активності

Moodle	Австралія	Вільно	IMS, SCORM	Так	Тести, завдання, семінари, активність на форумах, контроль активності
--------	-----------	--------	---------------	-----	---

Усі чотири розглянуті програмні продукти поширюються на основі GNU General Public License (GPL), яка дозволяє вільно використовувати, змінювати та доповнювати програму.

Обґрунтований вибір у певній системі дистанційного навчання зазвичай роблять за критеріями надійності та безпеки, сумісності, зручність у користуванні та адмініструванні, модульності, вартості програмного забезпечення, супроводу та апаратної частини.

Проведений аналіз поширених систем дистанційного навчання не дає підстав для здійснення однозначного вибору. Безкоштовні системи (з відкритим кодом) складні в обслуговуванні і потребують фахівців для впровадження та підтримки системи. Тому відповідь на вибір найраціональнішої системи є суб'єктивною та вимагає визначення чіткого переліку завдань та апаратних і фінансових можливостей навчального закладу.

## **2.2. Актуальність використання дистанційного навчання**

Розвиток високих технологій актуалізує проблему модернізації сучасної системи освіти. Суть такої модернізації найбільше відображає концепція дистанційної освіти. Сучасне суспільство вимагає постійного оновлення та вдосконалення наявного багажу знань; безупинного розширення власного сприйняття комплексності світу та адаптації до діяльності в інформаційному середовищі; пошуку балансу між загальнолюдськими цінностями та неминучою комерціалізацією освітніх

послуг. Виконати соціальне замовлення суспільства через збільшення асигнувань на освіту, збільшення кількості навчальних закладів та іншими традиційними способами не в змозі навіть заможні країни. Тому поява дистанційної освіти не випадкова, це закономірний етап розвитку та адаптації освіти до сучасних умов.

Впровадження системи дистанційного навчання у вищі навчальні заклади України продиктовано вимогами часу. Більшість ВНЗ Європи та США ввели таку зручну для студентів форму освіти набагато раніше, ніж Україна.

Чим же відрізняється дистанційна освіта від інших видів отримання знань і професії? Дистанційна форма навчання дає можливість гармонійно поєднувати навчання з повсякденним життям. На сьогодні – це доступна можливість одержати освіту з мінімальними фінансовими витратами при великому виборі спеціальностей. Така форма навчання дає свободу вибору місця, часу та темпу навчання. Крім того – це відкрита система навчання, що передбачає активне спілкування між викладачем і студентом, та студентів між собою за допомогою сучасних технологій і мультимедіа.

У багатьох закладах вдало поєднують традиційні форми навчання з технологіями системи управління навчанням «Moodle».

Студентам пропонується в процесі вивчення курсів можливість переглядати презентації за окремими темами навчальних дисциплін, проходити тестування, самостійно опрацьовувати додатковий теоретичний матеріал, виконувати завдання для самостійної роботи, обговорювати дискусійні питання на форумах тощо. Використання інтерактивних форм навчання, таких як виконання завдань у групах, проведення семінарів та дискусій у режимі он-лайн, використання веб-квестів, викликає зацікавленість, надає практичної спрямованості та підвищує ефективність засвоєння курсу.

Використання дистанційної форми навчання забезпечує гнучкість – можливість засвоєння матеріалу курсу в зручному темпі, з урахуванням підготовки, здібностей студентів. Це досягається наповненням курсу додатковими елементами або ресурсами для одержання більш детальної або додаткової інформації з незрозумілих тем, а також низки питань – підказок тощо. Система забезпечує можливість упровадження новітніх педагогічних, психологічних, методичних розробок. Зручним є відсутність обмежень у часі для засвоєння матеріалу; можливість самостійно опрацювати пропущений матеріал. Ще однією перевагою є модульність – розбиття матеріалу на окремі функціонально завершені теми, які вивчаються у міру засвоєння і відповідають здібностям окремого студента або групи загалом. Завдяки розміщенню електронних навчальних матеріалів та мультидоступу до них, студенти отримують можливість одночасного використання великого обсягу навчальної інформації, що підвищує економічну ефективність дистанційного методу навчання. Система дистанційного навчання має більші можливості контролю якості навчання, які передбачають використання самоконтролю, відсутність психологічних бар'єрів. Звітність та управління оцінками викладачем дає змогу статистичної обробки результатів та аналізу інформації що дозволяє диференціювати завдання за ступенем складності у відповідності до рівня підготовки студентів.

Незважаючи на досить об'ємний перелік позитивних якостей дистанційної освіти, як і в будь-якій іншій формі навчання, в ній можна виділити кілька недоліків. Перш за все, це ускладнена ідентифікація студентів, оскільки на сучасному етапі розвитку технологій перевірити, хто ж саме проходить тести або надсилає відповіді на завдання досить складно. З метою вирішення проблеми ми проводимо, по можливості, тестування та інші види опитування в системі управління навчанням



«Moodle» під час аудиторних занять; регламентуємо час на виконання окремих видів завдань тощо.

Дистанційне навчання потребує сильної мотивації й самоорганізації, вміння працювати самостійно. З метою подолання цієї проблеми ми пропонуємо використання різних форм активного спілкування між студентами групи і викладачем, проведення дискусій, чатів, що значно посилює мотивацію до навчання, поліпшує засвоєння матеріалу.

Потенціал дистанційних технологій оцінюється високо, значно підвищується адаптованість у професійній сфері. Дистанційна освіта вже зайняла одне із провідних місць у системі вищої освіти, але найбільший ефект досягається завдяки комплексному поєднанню традиційних форм та методів навчання з можливостями дистанційного навчання.

### 3. ТЕОРЕТИЧНА ЧАСТИНА

#### 3.1. Огляд матеріалу за темою роботи

Курс теорії програмування оснований на творчому синтезі вибраних формальних теорій (лямбда-вирахування, комбінаторна логіка, теорія категорій та ін.) та унікальної технологічної платформи Microsoft.NET, що забезпечує практично прозору інтеграцію коду на мовах програмування різних типів.

«Теорію програмування» умовно можна поділити на 2 частини. Перша частина присвячена основам функціонального підходу до програмування на прикладі мови LISP. Саме це і розглянемо в теорії програмування. Друга частина охоплює основні розділи програмування в застосуванні до об'єктно орієнтованих мов програмування (мова C#), більш детально розгляните в курсі «Технології програмування». Базові аспекти професіонального програмування викладаються в порівнянні мов C# і LISP на єдиній платформі Microsoft.NET. Це дає можливість більш глибоко проникнути в суть процесу створення програмних систем.

Розглянемо класифікацію мов і підходів до програмування.

Табл.1 Класифікація мов і підходів до програмування

Мови і підходи до програмування	Час виникнення	Переваги	Недоліки
Перші мови програмування. Ранні неструктурні підходи.	30-ті; 40-ві рр. XX ст.	Програми – лінійні послідовності елементарних операцій з регістрами в яких зберігаються дані.	Оптимізовані під апаратну архітектуру конкретного комп'ютера.

<p>Мови «високого рівня» Імперативний підхід (ALGOL, COBOL, Fortran, APL → BPL→C, Pascal, Basic). Програма – це набір директив, звернених до комп'ютера.</p>	<p>50-ті рр. XX ст.</p>	<p>1) абстрагування від конкретних деталей апаратного забезпечення. 2) повторне виконання програмних блоків – функції або процедур. 3) поява спеціалізованих програм (трансляторів), що перетворюють інструкції мови в коди машини.</p>	<p>1) вивчення певних мов потребувало багато часу і нових засобів.</p>
<p>Декларативний підхід (SML, LISP, interlisp, Common, Lisp, Haskell, Prolog). Програма – це набір команд, а не опис дій для виконання  -функціональний (LISP). Програма – це функція однієї або декількох змінних.</p>	<p>60-ті рр. XX ст</p>	<p>1) формалізується математичними засобами, а отже, легко тестувати і верифікувати програми 2) висока степінь абстракції  1) можливість обробки різноманітних даних. 2) автоматичний динамічний розподіл пам'яті для збереження даних. 3) робота програміста зосереджена на області дослідження повністю, а не рутинній роботі (правильність представлення даних, «зборка мусора»).</p>	<p>1) труднощі ефективної реалізації трансляторів  1) нелінійна структурна програма 2) відносно невисока ефективність реалізації.</p>
	<p>70-ті роки. XX ст.</p>	<p>1) високий рівень</p>	

-логічний підхід (Prolog, mercury). Програма – це сукупність правил або логічних висловлювань (проекти штучного інтелекту)		машинної незалежності. 2)можливість відкатів – повернення до попередньої підмети при негативному результаті одного з варіантів розв’язку	1)специфікація класу розв’язуваної задачі. 2)складність ефективної реалізації для прийняття рішень в реальному часі (для систем життєвого забезпечення)
Об’єктно - орієнтований підхід (C++, Visual Basic, Oberon). Програма – це опис об’єктів, їх властивостей, класів, спосіб їх взаємодії і операції над об’єктами.	80-ті pp. XX ст.	1) близькість до предметної області довільної структури і призначення, а отже, побудова складних моделей з динамічними властивостями 2) використання раніше розроблених бібліотек об’єктів і методів 3) програмне забезпечення універсальне за рахунок поліморфізмів об’єктів, класів і методів.	1) складність тестування (перевірка правильності) та верифікації (відповідність стандарту).

Функціональний підхід дає можливість прозоро моделювати текст програми математичними засобам, а отже, досить цікавий з теоретичної точки зору.

Складні програми при такому підході будуються шляхом агрегування функцій. При цьому текст програми представляє собою функцію, деякі аргументи якої можна розглядати теж як функції. Більш

того, типи окремих функцій, можуть бути змінними. Таким чином, забезпечується обробка різнородних даних [6].

**Множина** – сукупність об'єктів, які відрізняються за деякою ознакою.

Множини найчастіше записуються у вигляді:

а)  $A = \{a, b, c\}$  – перерахування конкретних елементів;

б)  $B = \{x | x > 0\}$  – опис характеристичної властивості елементів множини.

Порожню множину будемо позначати через  $\emptyset$  або  $\{\}$ , універсальну множину – через  $U$ . Універсальна множина містить всі елементи деякої предметної області. Наприклад, у теорії чисел це множина усіх цілих або натуральних чисел, у геометрії – множина усіх точок  $n$ -вимірного геометричного простору.

Відношення між множинами і елементами множин будемо записувати у вигляді:

- $a \in A$  –  $a$  належить  $A$ ;
  - $B \subseteq C$  –  $B$  є підмножиною  $C$ ,  $B$  включається (включене) у  $C$ ;
  - $B \subset C$  –  $B$  є власною підмножиною  $C$ ,  $B$  строго включається в  $C$ .
- $\emptyset$  і  $C$  вважаються невластими підмножинами множини  $C$ .

Множину всіх підмножин множини  $A$  називають булеаном множини  $A$  і позначають  $P(A)$ .

Операції над множинами.

Об'єднання:  $A \cup B = \{x | x \in A \text{ або } x \in B\}.$

Перетинання:  $A \cap B = \{x | x \in A \text{ і } x \in B\}$

Доповнення:  $\overline{A} = \{x | x \in U \text{ і } x \notin A\}$

Різниця:  $A \setminus B = \{x | x \in A \text{ і } x \notin B\}.$

Симетрична різниця:  $A \oplus B = \{x | x \in (A \setminus B) \text{ або } x \in (B \setminus A)\}.$

**Декартовим добутком**  $A \times B$  множин  $A$  і  $B$  називається множина усіх пар вигляду  $(a_i, b_j)$ , де  $a_i \in A, b_j \in B$

**Декартів добуток**  $A_1 \times A_2 \times \dots \times A_n$  множин  $A_1, A_2, \dots, A_n$  є множина усіх  $n$ -к вигляду  $(a_{1i}, a_{2j}, \dots, a_{nk})$ , де  $a_{1i} \in A_1, \dots, a_{nk} \in A_n$ .

Якщо  $A_1 = A_2 = \dots = A_n = A$ , декартів добуток називають **ступенем**  $A^n$ . При цьому  $A^0 = \emptyset$ ,  $A^1 = A$ .

Приклади.

$$1 \quad A = \{a, b\}; B = \{c, d\}; A \times B = \{(a, c), (b, c), (a, d), (b, d)\}$$

$$2 \quad A - \text{відрізок на осі } x, \quad B - \text{відрізок на осі } y.$$

$$A \times B - \text{прямокутник на площині } xy.$$

$$3 \quad X - \text{множина координат } x, \quad Y - \text{множина координат } y, \quad s - \text{множина довжин}, \quad a - \text{множина кутів}.$$

$$X \times Y \times s \times a - \text{множина усіх векторів на площині}.$$

Елементи множин  $A$  і  $B$  знаходяться у **взаємо-однозначній відповідності**, якщо кожному  $a \in A$  за деяким законом зіставлений один і той самий елемент  $b \in B$ , причому кожен  $b \in B$  виявляється зіставленим одному і тільки одному  $a \in A$ .

Множини  $A$  і  $B$  **рівнопотужні** (або еквівалентні), якщо можна установити взаємооднозначну відповідність їхніх елементів.

Приклади: множина  $N = \{1, 2, 3, \dots\}$  натуральних чисел еквівалентна множині  $N^{(2)} = \{1, 4, 9, \dots\}$  квадратів натуральних чисел, множина цілих чисел еквівалентна множині парних чисел.

Таким чином, нескінченна множина може бути еквівалентною своїй власній підмножині.

Множини, еквівалентні множині натуральних чисел  $N$ , називаються **зліченими**.

Об'єднання скінченної або зліченної сукупності злічених множин також є зліченною множиною.

Приклад. Множина усіх слів скінченної довжини в алфавіті  $a_1, a_2, \dots, a_m$  – зліченна.

Множина усіх дійсних чисел інтервалу  $(0,1)$  незліченна.

Множина, еквівалентна множині всіх дійсних чисел інтервалу  $(0,1)$ , називається множиною потужності **континуума**.

Множина усіх підмножин деякої зліченної множини є множиною потужності континуума.

Нехай  $F(A)$  – множина усіх слів у скінченному алфавіті  $A$ . Будь-яка підмножина  $L \subseteq F(A)$  називається мовою над алфавітом  $A$ .

Множина усіх мов над скінченним алфавітом  $A$  є множиною потужності континуума.

Потужність множини задається кардинальним числом. Кардинальне число зліченної множини позначають через  $\aleph_0$ .

Множини найбільшої потужності не існує.

Візьмемо довільні множини  $A_1, A_2, \dots, A_n$  не обов'язково різні.

Під  **$n$ -арним**, або  **$n$ -відношенням**, на множинах  $A_1, A_2, \dots, A_n$  розуміють закон, що виділяє у декартовому добутку деяку підмножину

$$\rho^n \subseteq A_1 \times A_2 \times \dots \times A_n.$$

Часто під відношенням розуміють саму підмножину  $\rho^n$ .

Відношення будемо позначати грецькими буквами  $\rho, \alpha, \beta, \gamma, \dots$  і спеціальними символами  $<, =, >, \leq, \geq$ .

Відношення  $\rho^n$  можна представити у матричній формі:

$$\rho^n = \left\| \begin{array}{ccc} a_1^{(1)} & . & a_1^{(n)} \\ . & . & . \\ a_r^{(1)} & . & a_r^{(n)} \end{array} \right\|.$$

Рядки матриці при цьому називають векторами відношення.

$\rho^1$  на  $A$  – унарне відношення,  $\rho^2$  на  $A, B$  – бінарне,  $\rho^3$  на  $A, B, C$  – тернарне.

#### *Унарні відношення*

Унарне відношення  $\rho^1$  на множині  $A$  є характеристичною властивістю деякої підмножини.

Приклади відношень на  $N$  :  $\alpha^1 = \{3, 7\}$ ;  $\beta^1 = \{n \mid n > 5\}$  ;  
 $\gamma^1 = \{n \mid n - \text{парне}\}$ .

Множина всіх унарних відношень на  $A$  збігається з множиною всіх підмножин  $A$ .

#### *Бінарні відношення*

Якщо  $a \in A$  і  $b \in B$  знаходяться у відношенні  $\rho$ , це іноді записують так:  $a\rho b$ .

Бінарні відношення часто представляють за допомогою таблиць.

Нехай є відношення  $\rho$  на  $A = \{a_1, \dots, a_r\}$  і  $B = \{b_1, \dots, b_s\}$ . Рядки таблиці відповідають елементам  $a_i$ , стовпці – елементам  $b_j$ , наявність відношення  $a_i \rho b_j$  позначається як “\*”.

$\rho$	2	3	5
2	*		*
3		*	

#### *Тернарне відношення*

Наприклад, з кожною бінарною операцією  $F(x, y)$ , зокрема “+”, “–”, можна зв'язати тернарне відношення  $\varphi^3(x, y, z)$ , для якого  $z = F(x, y)$ .

#### *Приклади n-арних відношень*

1.  $A = (2, 3)$ ,  $B = \{2, 3, 4, 6\}$ ,  $a$  і  $b$  знаходяться у відношенні  $\rho$ , якщо  $a$  є дільник  $b$ .



$$\rho = \begin{pmatrix} 2 & 2 \\ 2 & 4 \\ 2 & 6 \\ 3 & 3 \\ 3 & 6 \end{pmatrix}$$

2. Бінарне відношення  $<$  на  $N$ :

$$<_N = \begin{pmatrix} 1 & 2 \\ 1 & 3 \\ \cdot & \cdot \\ 2 & 3 \\ 2 & 4 \\ \cdot & \cdot \end{pmatrix}$$

3. Відношення  $<, =, >$  на множині слів.

4. Відношення  $\psi$ , що зіставляє кожному слову з  $F(A)$  його код  $q \in M$ . Тут  $A$  – алфавіт;  $M$  – деяка множина [7].

Відношення широко використовуються при описі синтаксису мов і при перекладі з однієї мови на іншу.

$$\text{Нехай знову } \rho^n = \begin{pmatrix} a_1^{(1)} & \cdot & a_1^{(n)} \\ \cdot & \cdot & \cdot \\ a_r^{(1)} & \cdot & a_r^{(n)} \end{pmatrix}.$$

**Проекцією**  $Pr^{(j)}(\rho^n)$  відношення  $\rho^n$  на множину  $A_j$  називається множина всіх елементів  $j$ -го стовпця матриці.

**Перетином**  $S^{(j)}(\rho^n)$  відношення  $\rho^n$  за елементами  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$  називається множина всіх елементів  $a_{i_q}^{(j)}$ , для яких  $n$ -ка  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_{i_q}^{(j)}, a_i^{(j+1)}, \dots, a_i^{(n)}$  належить даному відношенню.

**Фактор-множиною**  $A_j / \rho^n$  множини  $A_j$  стосовно  $\rho^n$  називається множина перетинів цього відношення за різними сукупностями  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$  [6].

Приклад. Нехай відношення  $\alpha^3$  на множинах  $A = \{a_1, a_2\}$ ,  $B = \{b_1, b_2, b_3\}$ ,  $C = \{0, 1\}$  визначається рівністю

$$\alpha^3 = \left\| \begin{array}{ccc} a_1 & b_1 & 0 \\ a_1 & b_2 & 0 \\ a_2 & b_1 & 1 \\ a_2 & b_2 & 1 \\ a_2 & b_2 & 0 \end{array} \right\|.$$

Тоді

$$\begin{aligned} \text{Пр}^1(\alpha^3) &= A, \text{Пр}^2(\alpha^3) = \{b_1, b_2\}, \text{Пр}^3(\alpha^3) = C; & S_{b_2 0}^1(a^3) \\ &= \{a_1, a_2\}, S_{a_1 0}^2(a^3) = \{b_1, b_2\}. \end{aligned}$$

Фактор-множина  $C / \alpha^3$  подається у вигляді

$$\{S_{a_1 b_1}^3(a^3) = \{0\}, S_{a_1 b_2}^3 = \{1\}, S_{a_2 b_2}^3 = C, S_{a_1 b_3}^3 = \emptyset, S_{a_2 b_3}^3 = \emptyset\}.$$

Відношення  $\varphi^n$  на множинах  $A_1, A_2, \dots, A_n$  називається функціональним, якщо для кожної послідовності  $a_i^{(1)}, \dots, a_i^{(n-1)} \in A_1 \times \dots \times A_{n-1}$  перетин  $S^n$  за цими елементами містить не більше ніж один елемент  $a_i^{(n)} \in A_n$ .

Якщо перетин містить точно один елемент  $a_i^{(n)}$ , то  $\varphi^n$  називають всюди визначеним функціональним відношенням.

Приклади. Нехай є множини  $A = \{a_1, a_2\}$ ,  $B = \{b_1, b_2\}$ ,  $C = \{0, 1\}$ . Відношення

$$\alpha^3 = \begin{vmatrix} a_1 & b_1 & 0 \\ a_1 & b_2 & 0 \\ a_2 & b_1 & 1 \end{vmatrix}, \quad \beta^3 = \begin{vmatrix} a_1 & b_1 & 1 \\ a_1 & b_2 & 0 \\ a_2 & b_1 & 0 \\ a_2 & b_2 & 1 \end{vmatrix}$$

– функціональні, причому  $\beta^3$  – усюди визначене.

З функціональним відношенням  $\varphi^n$  зв'язують функцію  $F(x_1, \dots, x_{n-1})$ . У вищенаведеному прикладі з  $\alpha^3$  зв'язана часткова функція  $F(x, y)$ :

$F(a_1, b_1) = F(a_1, b_2) = 0$ ,  $F(a_2, b_1) = 1$ , а з  $\beta^3$  зв'язана функція  $F(x, y)$ :

$$F(a_1, b_1) = F(a_2, b_2) = 1, \quad F(a_1, b_2) = F(a_2, b_1) = 0.$$

Нехай  $\varphi$  – бінарне відношення на множинах  $A, B$ .

Відношення  $\varphi$  називається відображенням множини  $A$  в  $B$ , якщо  $\varphi$  функціональне й усюди визначене.

Відображення  $\varphi$  множини  $A$  у  $B$  називається відображенням  $A$  на  $B$ , якщо кожний елемент  $b \in B$  має в  $A$  хоча б один прообраз.

Приклад. Є множини  $A = \{a, b, c, d\}$  і  $B = \{1, 2, 3\}$ .

Відношення  $\alpha = \begin{vmatrix} a & 1 \\ b & 1 \\ c & 2 \\ d & 3 \end{vmatrix}$  є відображенням  $A$  на  $B$ .

Бінарне відношення  $\beta = \begin{vmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 2 \end{vmatrix}$  є відображенням  $B$  у себе.

Бінарне відношення  $\rho$  на множині  $A$ :

- рефлексивне, якщо  $a\rho a$ ;
- транзитивне, якщо з  $a\rho b$  і  $b\rho c$  випливає  $a\rho c$ ;

- симетричне, якщо з  $a\rho b$  випливає  $b\rho a$ ;
- антисиметричне, якщо з  $a\rho b$  і  $b\rho a$  випливає  $a = b$ .

Бінарне відношення, що задовольняє властивості рефлексивності, транзитивності і симетричності, називається відношенням **еквівалентності**.

Кожному відношенню еквівалентності на множині  $A$  відповідає єдине розбиття даної множини на суміжні класи.

Бінарне відношення  $\rho$  на якійсь множині  $A$ , яке задовольняє властивості рефлексивності, транзитивності й антисиметричності, називається відношенням **часткового порядку** [7].

Приклади відношень часткового порядку:

1.  $\leq$  на деякій числовій множині;
2.  $\leq$  на множині слів скінченної довжини;  $s_1 \leq s_2$ , якщо  $|s_1| \leq |s_2|$ ;
3.  $\subseteq$  на множині слів;  $s_1 \subseteq s_2$ , якщо  $s_1$  є підсловом  $s_2$ ;
4.  $\subseteq$  відношення включення для підмножин деякої множини.

Відношення порядку символічно позначається через  $\leq$ .

Множина  $A$  з визначеним на ньому відношенням часткового порядку називається **частково упорядкованою**.

Елементи  $a, b \in A$  такі, що  $a\rho b$  або  $b\rho a$  називаються порівнянними елементами частково упорядкованої множини  $A$ .

Так, у прикладах 3, 4 є непорівнянні елементи.

Частково упорядкована множина, у якій кожна пара елементів порівнянна, називається **лінійно упорядкованою множиною**, або **ланцюгом**.

Нехай  $A$  – частково упорядкована множина;  $B$  – його підмножина. **Верхньою гранню**  $B$  у множині  $A$  називається елемент  $a \in A$ , такий, що  $b \leq a$  для будь-якого  $b \in B$ . Елемент  $a$  – **найбільший** у множині  $A$ , якщо

$a$  – верхня грань самого  $A$ . Елемент  $m \in A$  називається **максимальним** у множині  $A$ , якщо не існує  $b \in A$ , такого, що  $m < b$ .

Найбільший елемент множини  $A$  є єдиним максимальним елементом даної множини. Множина  $A$  може не мати найбільшого елемента і в той же час мати один або кілька максимальних.

Аналогічні міркування справедливі для нижньої грані.

Приклади.

1. Множина усіх підмножин множини  $A$  має найменший елемент  $\emptyset$  і найбільший елемент – саму  $A$ .
2. Множина  $Z$  усіх цілих чисел не має найменшого, найбільшого, мінімального і максимального елементів.
3. Множина  $N$  натуральних чисел має найменший елемент 1 і не має найбільшого.
4. Множина усіх непорожніх ланцюжків має мінімальні елементи – односимвольні ланцюжки – і не має найменшого, найбільшого і максимального.

Лінійно впорядкована множина  $A$  називається **повністю впорядкованою**, якщо кожна непорожня підмножина  $B \subseteq A$  має найменший елемент.

Множина  $N$  є повністю впорядкованою, множина  $Z$  – ні.

Нехай  $B$  – підмножина частково впорядкованої множини  $A$ .

Верхнім конусом  $B^\Delta$  множини  $B$  називається сукупність усіх верхніх граней даної множини, **точною верхньою гранню**  $B$  у множині  $A$  – елемент  $a \in B^\Delta$ , такий, що  $a \leq b$  для будь-якого  $b \in B$ .

Частково упорядкована множина  $S$  називається **структурою**, або решіткою, якщо для будь-якої пари елементів  $a, b \in S$  існує точна нижня грань і точна верхня грань [6].

Приклади структур

1. Множина  $N$ .

2. Ланцюг у частково упорядкованій множині з відношенням порядку.

3. Множина усіх підмножин деякої множини, частково упорядкована за включенням, де  $\inf \{A_1, A_2\} = A_1 \cap A_2$ ,  $\sup \{A_1, A_2\} = A_1 \cup A_2$ .

4. Структурою є множина усіх мов у даному алфавіті.

Частково упорядкована множина  $S$  називається **повною структурою**, якщо для будь-якої непорожньої підмножини  $B \subseteq S$  у множині  $S$  існують точні нижня і верхня грані.

Структури з прикладів 2 – 4 – повні, структура з прикладу 1 – неповна, тому що будь-яка нескінченна послідовність не має верхньої грані.

Точна нижня грань усієї множини  $S$  називається нулем повної структури, а точна верхня грань – одиницею даної структури.

Дистрибутивна структура  $S$  з елементами  $0$  і  $1$  називається **булевою структурою**, або булевою алгеброю, якщо для будь-якого елемента  $a \in S$  існує елемент  $\bar{a} \in S$  такий, що  $a \cap \bar{a} = 0$  і  $a \cup \bar{a} = 1$ . Елемент  $\bar{a}$  називається доповненням елемента  $a$  у структурі  $S$ .

Булевою структурою є множина усіх мов над алфавітом  $A$  [8].

### 3.2. Алгоритмізація задачі за темою роботи

Користувачу пропонується спочатку переглянути теоретичний матеріал за темою (передбачена можливість його завантажити). На кожному кроці тренажеру виводиться завдання, відповідь надається одним із способів:

- вибір одного з декількох варіантів;
- вибір декількох варіантів;

- встановлення відповідності.

При неправильній відповіді відображається повідомлення про помилку.

Всі завдання представлено з таких тем:

1. Множини;
2. Відношення;
3. Порядок, структури.

**Крок 1.** Виводиться завдання: Множина – це

Виберіть правильну відповідь:

- об'єкти, які відрізняються за деякою ознакою;
- сукупність об'єктів, які відрізняються за деякою ознакою;
- сукупність об'єктів, які схожі за деякою ознакою.

Якщо не вибрано другий варіант, то відображається помилка: «Множина – сукупність об'єктів, які відрізняються за деякою ознакою». Інакше – перехід на наступний крок.

**Крок 2.** Виводиться завдання: Множини найчастіше записуються у вигляді а)  $A = \{a, b, c\}$  –

Виберіть правильну відповідь:

- перерахування конкретних елементів;
- опис характеристичної властивості елементів множини.

Якщо не вибрано перший варіант, то відображається помилка: «Множини найчастіше записуються у вигляді а)  $A = \{a, b, c\}$  – перерахування конкретних елементів». Інакше – перехід на наступний крок.

**Крок 3.** Виводиться завдання: Множини найчастіше записуються у вигляді б)  $B = \{x \mid x > 0\}$  –

Виберіть правильну відповідь:

- перерахування конкретних елементів;
- опис характеристичної властивості елементів множини.

Якщо не вибрано перший варіант, то відображається помилка: «Множини найчастіше записуються у вигляді б)  $B = \{x | x > 0\}$  – опис характеристичної властивості елементів множини». Інакше – перехід на наступний крок.

**Крок 4.** Виводиться завдання: Порожню множину будемо позначати через

Виберіть правильні відповіді:

- $\emptyset$ ;
- $\{\}$ ;
- $U$ .

Якщо вибрано третій варіант, то відображається помилка: «Порожню множину будемо позначати через  $\emptyset$  або  $\{\}$  ». Інакше – перехід на наступний крок.

**Крок 5.** Виводиться завдання: Універсальну множину будемо позначати через

Виберіть правильну відповідь:

- $\emptyset$ ;
- $\{\}$ ;
- $U$ .

Якщо не вибрано третій варіант, то відображається помилка: «Універсальну множину будемо позначати через  $U$  ». Інакше – перехід на наступний крок.

**Крок 6.** Виводиться завдання: Відношення між множинами і елементами множин будемо записувати у вигляді

Встановіть відповідність:

$a \in A$  –  $a$  належить  $A$ ;

$B \subseteq C$  –  $B$  є власною підмножиною  $C$ ,  $B$  строго включається в  $C$ ;

$B \subset C$  –  $B$  є підмножиною  $C$ ,  $B$  включається (включене) у  $C$ .



Якщо неправильно встановлено відповідність, то відображається помилка: « $a \in A$  –  $a$  належить  $A$ ;

$B \subseteq C$  –  $B$  є підмножиною  $C$ ,  $B$  включається (включене) у  $C$ ;

$B \subset C$  –  $B$  є власною підмножиною  $C$ ,  $B$  строго включається в  $C$ ».

Інакше – перехід на наступний крок.

**Крок 7.** Виводиться завдання: Операції над множинами

Встановіть відповідність:

Об'єднання:  $\overline{A} = \{x \mid x \in U \text{ і } x \notin A\}$ .

Перетинання:  $A \cup B = \{x \mid x \in A \text{ або } x \in B\}$

Доповнення:  $A \cap B = \{x \mid x \in A \text{ і } x \in B\}$

Різниця:  $A \oplus B = \{x \mid x \in (A \setminus B) \text{ або } x \in (B \setminus A)\}$ .

Симетрична різниця:  $A \setminus B = \{x \mid x \in A \text{ і } x \notin B\}$ .

Якщо неправильно встановлено відповідність, то відображається помилка: «Об'єднання:  $A \cup B = \{x \mid x \in A \text{ або } x \in B\}$ .

Перетинання:  $A \cap B = \{x \mid x \in A \text{ і } x \in B\}$

Доповнення:  $\overline{A} = \{x \mid x \in U \text{ і } x \notin A\}$

Різниця:  $A \setminus B = \{x \mid x \in A \text{ і } x \notin B\}$ .

Симетрична різниця:  $A \oplus B = \{x \mid x \in (A \setminus B) \text{ або } x \in (B \setminus A)\}$  .».

Інакше – перехід на наступний крок.

**Крок 8.** Виводиться завдання: Декартовим добутком  $A \times B$  множин  $A$  і  $B$  називається

Виберіть правильну відповідь:

- множина усіх пар вигляду  $(a_i, b_j)$ , де  $a_i \in A, b_j \in B$ ;
- множина усіх  $n$ -к вигляду  $(a_{1i}, a_{2j}, \dots, a_{nk})$ , де  $a_{1i} \in A_1, \dots, a_{nk} \in A_n$ ;
- обидві відповіді правильні.

Якщо не вибрано перший варіант, то відображається помилка: «Декартовим добутком  $A \times B$  множин  $A$  і  $B$  називається множина усіх пар вигляду  $(a_i, b_j)$ , де  $a_i \in A, b_j \in B$ ». Інакше – перехід на наступний крок.

**Крок 9.** Виводиться завдання: Декартів добуток  $A_1 \times A_2 \times \dots \times A_n$  множин  $A_1, A_2, \dots, A_n \in$

Виберіть правильну відповідь:

- множина усіх пар вигляду  $(a_i, b_j)$ , де  $a_i \in A, b_j \in B$ ;
- множина усіх  $n$ -к вигляду  $(a_{1i}, a_{2j}, \dots, a_{nk})$ , де  $a_{1i} \in A_1, \dots, a_{nk} \in A_n$ ;
- обидві відповіді правильні.

Якщо не вибрано другий варіант, то відображається помилка: «Декартів добуток  $A_1 \times A_2 \times \dots \times A_n$  множин  $A_1, A_2, \dots, A_n \in$  множина усіх  $n$ -к вигляду  $(a_{1i}, a_{2j}, \dots, a_{nk})$ , де  $a_{1i} \in A_1, \dots, a_{nk} \in A_n$ ». Інакше – перехід на наступний крок.

**Крок 10.** Виводиться завдання: Декартів добуток називають ступенем  $A^n$ , якщо

Виберіть правильну відповідь:

- $A_1 = A_2 = \dots = A_n = A$ ;
- $A_1 \neq A_2 \neq \dots \neq A_n \neq A$ ;
- $A_1 = A_2 = \dots = A_{n-1} = A$ .

Якщо не вибрано перший варіант, то відображається помилка: «Якщо  $A_1 = A_2 = \dots = A_n = A$ , декартів добуток називають ступенем  $A^n$ ». Інакше – перехід на наступний крок.

**Крок 11.** Виводиться завдання: Елементи множин  $A$  і  $B$  знаходяться у взаємо-однозначній відповідності, якщо

Виберіть правильну відповідь:

- вони еквівалентні множині натуральних чисел  $N$ ;

- можна установити взаємооднозначну відповідність їхніх елементів;
- кожному  $a \in A$  за деяким законом зіставлений один і той самий елемент  $b \in B$ , причому кожен  $b \in B$  виявляється зіставленим одному і тільки одному  $a \in A$ .

Якщо не вибрано третій варіант, то відображається помилка: «Елементи множин  $A$  і  $B$  знаходяться у взаємо-однозначній відповідності, якщо кожному  $a \in A$  за деяким законом зіставлений один і той самий елемент  $b \in B$ , причому кожен  $b \in B$  виявляється зіставленим одному і тільки одному  $a \in A$ ». Інакше – перехід на наступний крок.

**Крок 12.** Виводиться завдання: Множини  $A$  і  $B$  рівнопотужні (або еквівалентні), якщо

Виберіть правильну відповідь:

- вони еквівалентні множині натуральних чисел  $N$ ;
- можна установити взаємооднозначну відповідність їхніх елементів;
- кожному  $a \in A$  за деяким законом зіставлений один і той самий елемент  $b \in B$ , причому кожен  $b \in B$  виявляється зіставленим одному і тільки одному  $a \in A$ .

Якщо не вибрано другий варіант, то відображається помилка: «Множини  $A$  і  $B$  рівнопотужні (або еквівалентні), якщо можна установити взаємооднозначну відповідність їхніх елементів». Інакше – перехід на наступний крок.

**Крок 13.** Виводиться завдання: Множини називаються зліченими, якщо

Виберіть правильну відповідь:

- вони еквівалентні множині натуральних чисел  $N$ ;
- можна установити взаємооднозначну відповідність їхніх елементів;

- кожному  $a \in A$  за деяким законом зіставлений один і той самий елемент  $b \in B$ , причому кожен  $b \in B$  виявляється зіставленим одному і тільки одному  $a \in A$ .

Якщо не вибрано перший варіант, то відображається помилка: «Множини, еквівалентні множині натуральних чисел  $N$ , називаються зліченими». Інакше – перехід на наступний крок.

**Крок 14.** Виводиться завдання: Візьмемо довільні множини  $A_1, A_2, \dots, A_n$  не обов'язково різні. Під  $n$ -арним, або  $n$ -відношенням, на множинах  $A_1, A_2, \dots, A_n$  розуміють закон, що виділяє у декартовому добутку деяку підмножину

Виберіть правильну відповідь:

- $\rho^n \subseteq A_1 + A_2 + \dots + A_n$ ;
- $\rho^n \subseteq A_1 \times A_2 \times \dots \times A_n$ ;
- $\rho^n \in A_1 \times A_2 \times \dots \times A_n$ .

Якщо не вибрано другий варіант, то відображається помилка: «Під  $n$ -арним, або  $n$ -відношенням, на множинах  $A_1, A_2, \dots, A_n$  розуміють закон, що виділяє у декартовому добутку деяку підмножину  $\rho^n \subseteq A_1 \times A_2 \times \dots \times A_n$ ». Інакше – перехід на наступний крок.

**Крок 15.** Виводиться завдання: Відношення  $\rho^n$  можна представити у матричній формі

Виберіть правильну відповідь:

- $\rho^n = \left\| \begin{array}{ccc} a_1^{(1)} & . & a_n^{(1)} \\ . & . & . \\ a_1^{(r)} & . & a_r^{(n)} \end{array} \right\|$ ;

$$\bullet \quad \rho^n = \left\| \begin{array}{ccc} a_1^{(1)} & . & a_1^{(n)} \\ . & . & . \\ a_r^{(1)} & . & a_r^{(n)} \end{array} \right\|.$$

Якщо не вибрано другий варіант, то відображається помилка:  
«Відношення  $\rho^n$  можна представити у матричній формі:

$$\rho^n = \left\| \begin{array}{ccc} a_1^{(1)} & . & a_1^{(n)} \\ . & . & . \\ a_r^{(1)} & . & a_r^{(n)} \end{array} \right\| \text{»}.$$

Інакше – перехід на наступний крок.

**Крок 16.** Виводиться завдання: Встановіть відповідність:

- $\rho^1$  на  $A$  – тернарне;
- $\rho^2$  на  $A, B$  – унарне відношення;
- $\rho^3$  на  $A, B, C$  – бінарне.

Якщо неправильно встановлено відповідність, то відображається помилка: « $\rho^1$  на  $A$  – унарне відношення;

- $\rho^2$  на  $A, B$  – бінарне;
  - $\rho^3$  на  $A, B, C$  – тернарне.».
- Інакше – перехід на наступний крок.

**Крок 17.** Виводиться завдання: Проекцією  $Pr^{(j)}(\rho^n)$  відношення

$\rho^n$  на множину  $A_j$  називається

Виберіть правильну відповідь:

- множина всіх елементів  $j$ -го стовпця матриці;
- множина всіх елементів  $a_{i_q}^{(j)}$ , для яких  $n$ -ка

$a_i^{(1)}, \dots, a_i^{(j-1)}, a_{i_q}^{(j)}, a_i^{(j+1)}, \dots, a_i^{(n)}$  належить даному відношенню;

- множина перетинів цього відношення за різними сукупностями  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$ .

Якщо не вибрано перший варіант, то відображається помилка: «Проекцією  $Pr^{(j)}(\rho^n)$  відношення  $\rho^n$  на множину  $A_j$  називається множина всіх елементів  $j$ -го стовпця матриці». Інакше – перехід на наступний крок.

**Крок 18.** Виводиться завдання: Перетином  $S^{(j)}(\rho^n)$  відношення  $\rho^n$  за елементами  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$  називається

Виберіть правильну відповідь:

- множина всіх елементів  $j$ -го стовпця матриці;
- множина всіх елементів  $a_{i_q}^{(j)}$ , для яких  $n$ -ка  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_{i_q}^{(j)}, a_i^{(j+1)}, \dots, a_i^{(n)}$  належить даному відношенню;
- множина перетинів цього відношення за різними сукупностями  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$ .

Якщо не вибрано другий варіант, то відображається помилка: «Перетином  $S^{(j)}(\rho^n)$  відношення  $\rho^n$  за елементами  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$  називається множина всіх елементів  $a_{i_q}^{(j)}$ , для яких  $n$ -ка  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_{i_q}^{(j)}, a_i^{(j+1)}, \dots, a_i^{(n)}$  належить даному відношенню». Інакше – перехід на наступний крок.

**Крок 19.** Виводиться завдання: Фактор-множиною  $A_j / \rho^n$  множини  $A_j$  стосовно  $\rho^n$  називається

Виберіть правильну відповідь:

- множина всіх елементів  $j$ -го стовпця матриці;
- множина всіх елементів  $a_{i_q}^{(j)}$ , для яких  $n$ -ка  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j)}, a_i^{(j+1)}, \dots, a_i^{(n)}$  належить даному відношенню;
- множина перетинів цього відношення за різними сукупностями  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$ .

Якщо не вибрано третій варіант, то відображається помилка: «Фактор-множиною  $A_j / \rho^n$  множини  $A_j$  стосовно  $\rho^n$  називається множина перетинів цього відношення за різними сукупностями  $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$ ». Інакше – перехід на наступний крок.

**Крок 20.** Виводиться завдання: Бінарне відношення  $\rho$  на множині  $A$  Встановіть відповідність:

рефлексивне,	якщо з $a\rho b$ і $b\rho c$ випливає $a\rho c$ ;
транзитивне,	якщо $a\rho a$ ;
симетричне,	якщо з $a\rho b$ і $b\rho a$ випливає $a = b$ ;
антисиметричне,	якщо з $a\rho b$ випливає $b\rho a$ .

Якщо неправильно встановлено відповідність, то відображається помилка: «рефлексивне, якщо  $a\rho a$ ;

транзитивне, якщо з  $a\rho b$  і  $b\rho c$  випливає  $a\rho c$ ;

симетричне, якщо з  $a\rho b$  випливає  $b\rho a$ ;

антисиметричне, якщо з  $a\rho b$  і  $b\rho a$  випливає  $a = b$ . ». Інакше – перехід на наступний крок.

**Крок 21.** Виводиться завдання: Бінарне відношення, що задовольняє властивості рефлексивності, транзитивності і симетричності, називається

Виберіть правильну відповідь:

- відношенням часткового порядку;

- відношенням еквівалентності.

Якщо не вибрано другий варіант, то відображається помилка: «Бінарне відношення, що задовольняє властивості рефлексивності, транзитивності і симетричності, називається відношенням еквівалентності». Інакше – перехід на наступний крок.

**Крок 22.** Виводиться завдання: Бінарне відношення  $\rho$  на якійсь множині  $A$ , яке задовольняє властивості рефлексивності, транзитивності й антисиметричності, називається

Виберіть правильну відповідь:

- відношенням часткового порядку;
- відношенням еквівалентності.

Якщо не вибрано перший варіант, то відображається помилка: «Бінарне відношення  $\rho$  на якійсь множині  $A$ , яке задовольняє властивості рефлексивності, транзитивності й антисиметричності, називається відношенням часткового порядку». Інакше – перехід на наступний крок.

**Крок 23.** Виводиться завдання: Множина  $A$  з визначенням на ньому відношенням часткового порядку називається

Виберіть правильну відповідь:

- частково упорядкованою;
- лінійно упорядкованою множиною, або ланцюгом.

Якщо не вибрано перший варіант, то відображається помилка: «Множина  $A$  з визначенням на ньому відношенням часткового порядку називається частково упорядкованою». Інакше – перехід на наступний крок.

**Крок 24.** Виводиться завдання: Частково упорядкована множина, у якій кожна пара елементів порівнянна, називається

Виберіть правильну відповідь:

- частково упорядкованою;



- лінійно упорядкованою множиною, або ланцюгом.

Якщо не вибрано другий варіант, то відображається помилка: «Частково упорядкована множина, у якій кожна пара елементів порівнянна, називається лінійно упорядкованою множиною, або ланцюгом». Інакше – перехід на наступний крок.

**Крок 25.** Виводиться завдання: Частково упорядкована множина  $S$  називається структурою, або решіткою, якщо

Виберіть правильну відповідь:

- для будь-якого елемента  $a \in S$  існує елемент  $\bar{a} \in S$  такий, що  $a \cap \bar{a} = 0$  і  $a \cup \bar{a} = 1$ . Елемент  $\bar{a}$  називається доповненням елемента  $a$  у структурі  $S$ ;
- для будь-якої пари елементів  $a, b \in S$  існує точна нижня грань і точна верхня грань;
- для будь-якої непорожньої підмножини  $B \subseteq S$  у множині  $S$  існують точні нижня і верхня грані.

Якщо не вибрано другий варіант, то відображається помилка: «Частково упорядкована множина  $S$  називається структурою, або решіткою, якщо для будь-якої пари елементів  $a, b \in S$  існує точна нижня грань і точна верхня грань». Інакше – перехід на наступний крок.

**Крок 26.** Виводиться завдання: Частково упорядкована множина  $S$  називається повною структурою, якщо

Виберіть правильну відповідь:

- для будь-якого елемента  $a \in S$  існує елемент  $\bar{a} \in S$  такий, що  $a \cap \bar{a} = 0$  і  $a \cup \bar{a} = 1$ . Елемент  $\bar{a}$  називається доповненням елемента  $a$  у структурі  $S$ ;
- для будь-якої пари елементів  $a, b \in S$  існує точна нижня грань і точна верхня грань;

- для будь-якої непорожньої підмножини  $B \subseteq S$  у множині  $S$  існують точні нижня і верхня грані.

Якщо не вибрано третій варіант, то відображається помилка: «Частково упорядкована множина  $S$  називається повною структурою, якщо для будь-якої непорожньої підмножини  $B \subseteq S$  у множині  $S$  існують точні нижня і верхня грані». Інакше – перехід на наступний крок.

**Крок 27.** Виводиться завдання: Дистрибутивна структура  $S$  з елементами  $0$  і  $1$  називається булевою структурою, або булевою алгеброю, якщо

Виберіть правильну відповідь:

- для будь-якого елемента  $a \in S$  існує елемент  $\bar{a} \in S$  такий, що  $a \cap \bar{a} = 0$  і  $a \cup \bar{a} = 1$ . Елемент  $\bar{a}$  називається доповненням елемента  $a$  у структурі  $S$ ;
- для будь-якої пари елементів  $a, b \in S$  існує точна нижня грань і точна верхня грань;
- для будь-якої непорожньої підмножини  $B \subseteq S$  у множині  $S$  існують точні нижня і верхня грані.

Якщо не вибрано перший варіант, то відображається помилка: «Дистрибутивна структура  $S$  з елементами  $0$  і  $1$  називається булевою структурою, або булевою алгеброю, якщо для будь-якого елемента  $a \in S$  існує елемент  $\bar{a} \in S$  такий, що  $a \cap \bar{a} = 0$  і  $a \cup \bar{a} = 1$ . Елемент  $\bar{a}$  називається доповненням елемента  $a$  у структурі  $S$ ». Інакше – перехід на наступний крок.

**Крок 28.** Виводиться результат проходження тренажеру. Пропонується перейти на головну сторінку.

### 3.3. Обґрунтування вибору програмних засобів для реалізації завдання роботи

Netlify - це компанія, що займається хмарними обчисленнями з Сан-Франциско, яка пропонує хостинг і бессерверні серверні служби для веб-додатків і статичних веб-сайтів.

Vue.js (читається як "в'ю", з англ. view) — JavaScript-фреймворк що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних, через реактивне зв'язування даних.

Vue використовує синтаксис шаблонів на основі HTML, що дозволяє декларативно зв'язувати рендеринг DOM з основними екземплярами даних в Vue. Всі Vue шаблони валідні HTML, і можуть бути розпарсені браузерами та HTML парсерами. Всередині Vue компілює шаблони в рендерингові функції віртуального DOM. В поєднанні з реактивною системою, Vue здатний розумно обчислити кількість компонентів для ре-рендингу та застосувати мінімальну кількість маніпуляцій з DOM, коли стан застосунку зміниться.

В Vue Ви можете використовувати синтаксис шаблонів або напямую писати рендерингові функції використовуючи JSX. Для того, щоб це зробити просто замініть шаблон на рендерингову функцію. Рендерингова функція відкриває можливості для потужних патернів базованих на компонентах - для прикладу, нова транзитна система тепер повністю базована на компонентах, що використовує рендерингові функції всередині.

Одна із найвиразніших особливостей Vue — це ненав'язлива реактивна система. Моделі це просто плоскі JavaScript об'єкти. Це робить керування станами дуже простим та інтуїтивним. Vue надає оптимізований ре-рендеринг з коробки без потреби робити що-небудь додатково. Кожен компонент слідує за своїми реактивними залежностями під час

рендерингу, тому система знає точно коли має відбуватись ре-рендеринг і які компоненти потрібно ре-рендерити.

Vue надає різноманітні шляхи для застосування ефектів переходу, коли елемент додають, оновлюють або видаляють з DOM.

Vue сам по собі не включає роутингу, та є vue-router пакет, який вирішує це питання. Він підтримує зв'язування вкладених шляхів з вкладеними компонентами і пропонує деталізований контроль над переходами. Vue дозволяє створення додатків за допомогою компонентів. Якщо додати vue-router до цього, все що потрібно зробити це зв'язати ваші компоненти з роутами і дозвольте vue-router вирішувати де їх рендерити [9].

Vue Router - офіційна бібліотека маршрутизації для Vue.js. Вона глибоко інтегрується з Vue.js і дозволяє легко створювати SPA-додатки. Включає наступні можливості [10]:

- Вкладені маршрути / уявлення
- Модульна конфігурація маршрутизатора
- Доступ до параметрів маршруту, query, wildcards
- Анімація переходів уявлень на основі Vue.js
- Зручний контроль навігації
- Автоматичне проставляння активного CSS класу для посилань
- Режими роботи HTML5 history або хеш, з авто-перемиканням в

IE9

- Що настраюється поведінку прокрутки сторінки

## 4. ПРАКТИЧНА ЧАСТИНА

### 4.1. Розробка блок-схеми, яка підлягає програмуванню

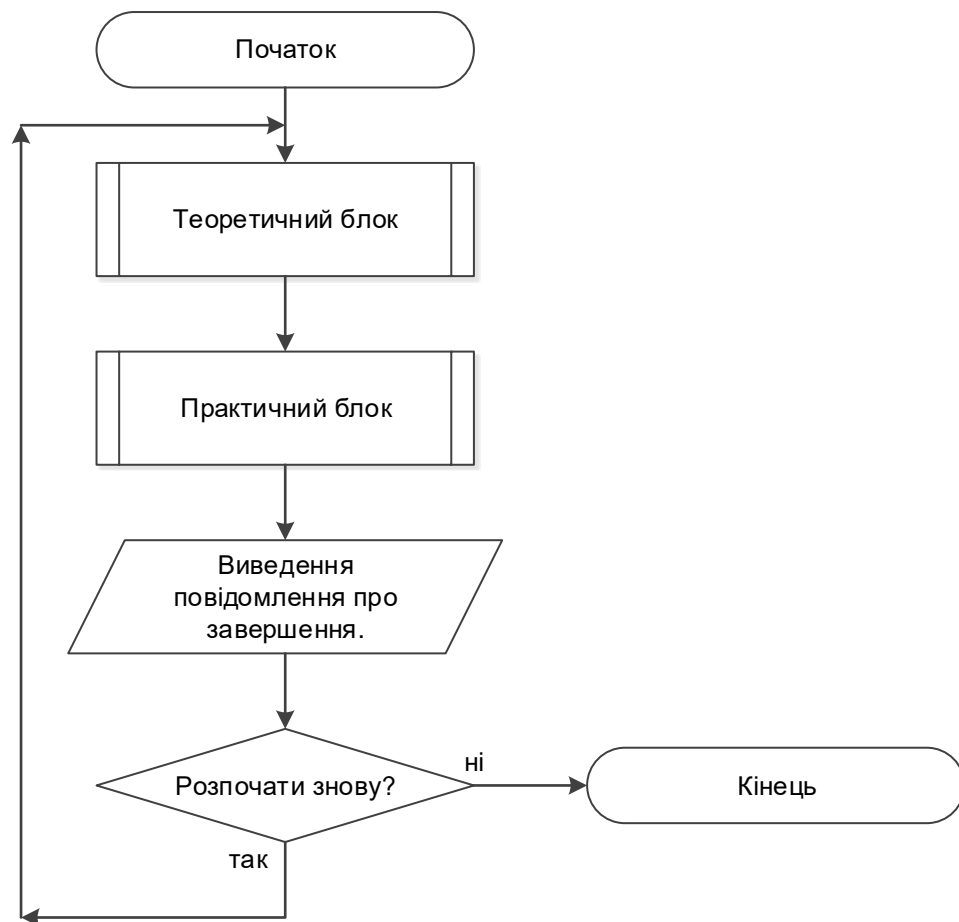


Рисунок 4.1 – Блок-схема роботи тренажеру

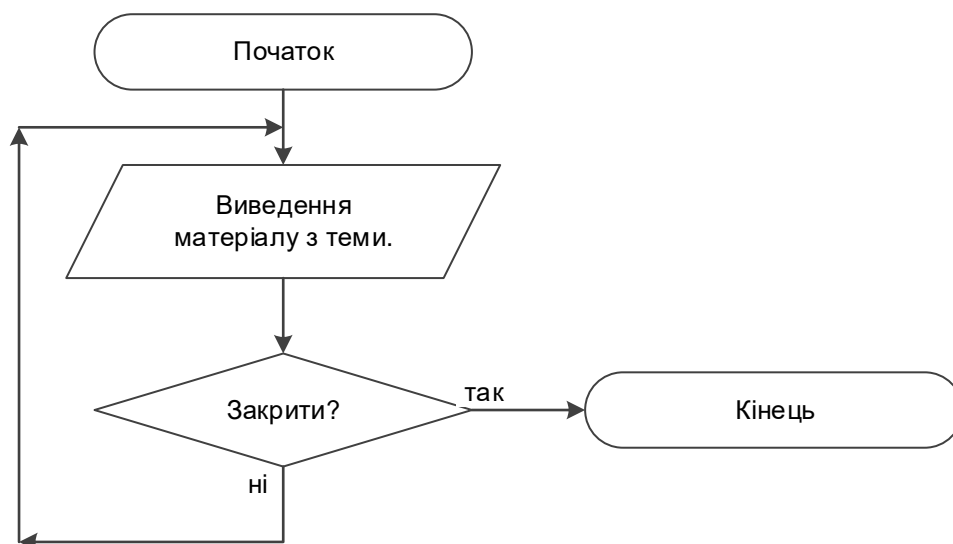


Рисунок 4.2 – Продовження блок-схеми роботи тренажеру

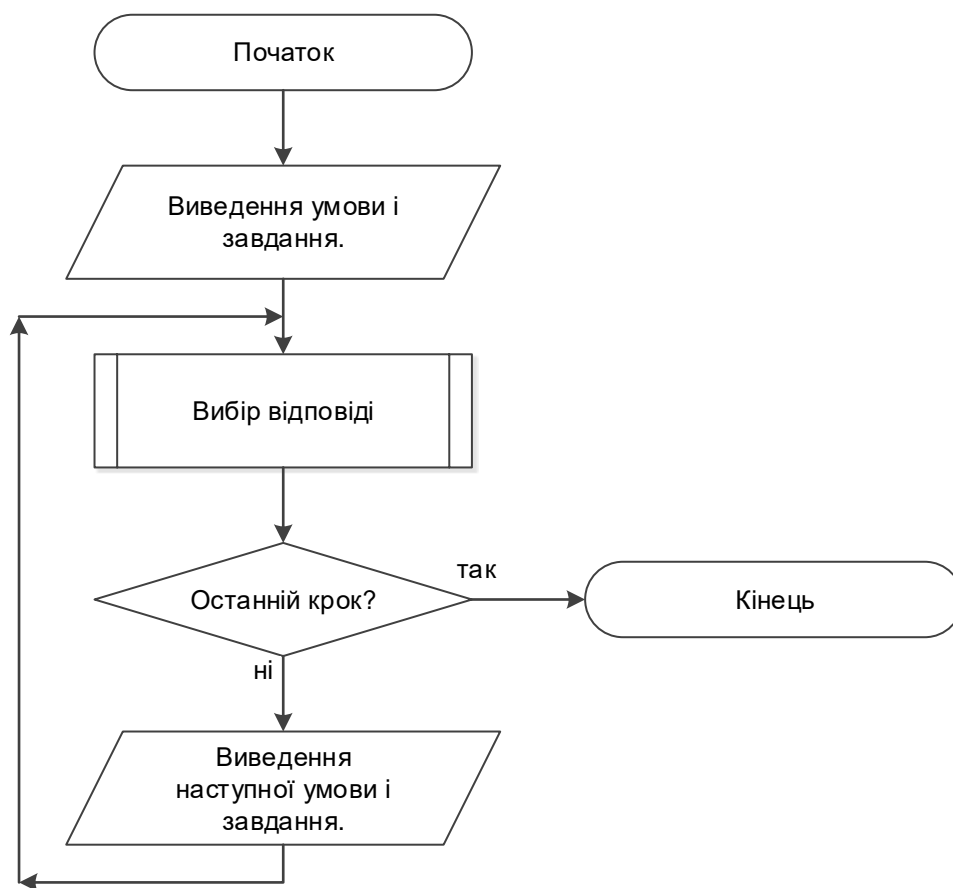


Рисунок 4.3 – Продовження блок-схеми роботи тренажеру

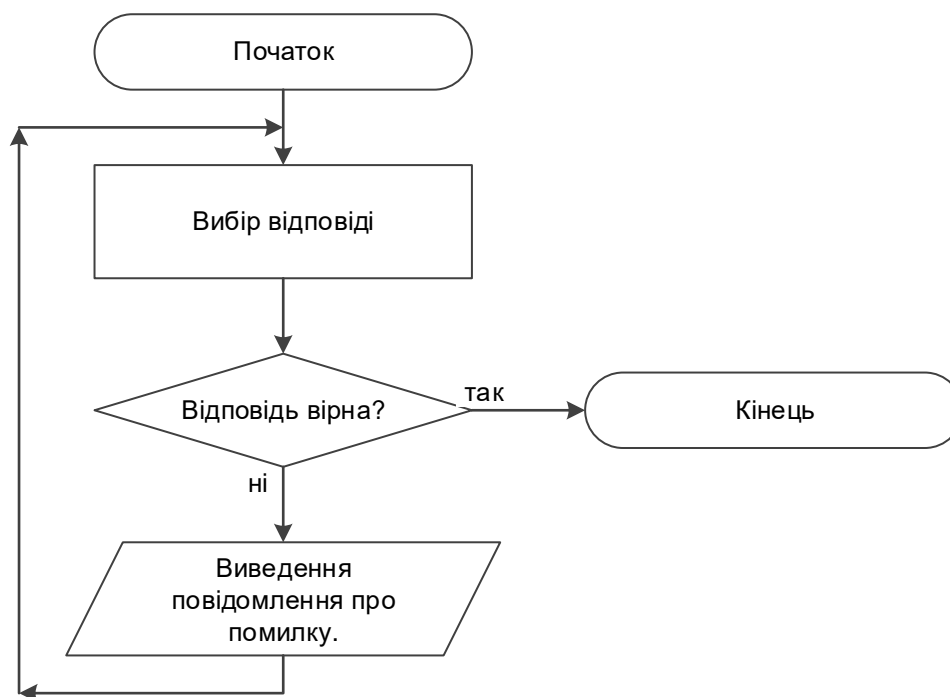


Рисунок 4.4 – Продовження блок-схеми роботи тренажеру

## 4.2. Опис процесу програмної реалізації

<https://diploma-puet.netlify.app> - тренажер завантажено на Netlify - безсерверну платформу, для створювання, розгортання та спільної роботи над веб-додатками.

При створенні програми за основу був взятий фреймворк Vue.js, в основі якого знаходиться javascript, для зручності навігації між сторінками була використана бібліотека vue-router.

Для стилізації компонентів був використаний UI-фреймворк Quasar. Для зручності стилізації замість звичайного CSS був обраний препроцесор SASS, який при білді компілюється та мініфікується в CSS.

Тренажер складається з трьох сторінок:

1. Початкова сторінка, на якій можна переглянути або завантажити матеріал дипломної роботи та розпочати тренування (рис. 4.5):

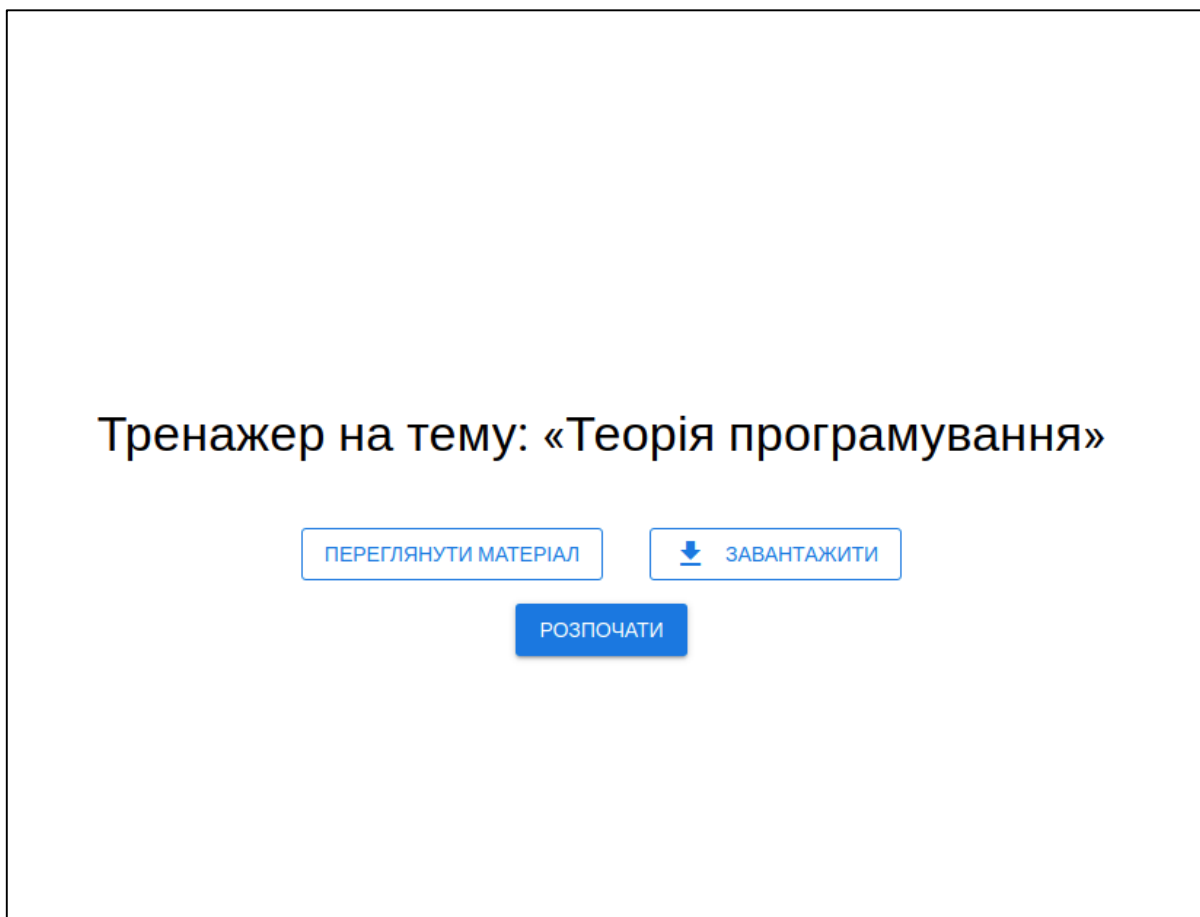


Рисунок 4.5 – Початкова сторінка

При натисканні на кнопку “РОЗПОЧАТИ”, користувач перейде на новий роут: “/quiz”

2. Сторінка із запитаннями (рис. 4.6).

Всі запитання знаходяться в простому JSON файлі та з легкістю можуть бути відредаговані чи доповнені.

Тренажер розрахований на запитання трьох типів:

- з однією правильною відповіддю - відображається radio group, в якій потрібно обрати правильну відповідь;
- з декількома правильними відповідями - відображаються checkboxes, з яких потрібно обрати правильні;
- встановлення відповідності - відображаються 2 вертикальні списки, які можна перетягувати за допомогою drag-n-drop.



**Множина – це**  
Виберіть правильну відповідь:

☐ об'єкти, які відрізняються за деякою ознакою;

☐ сукупність об'єктів, які відрізняються за деякою ознакою;

☒ сукупність об'єктів, які схожі за деякою ознакою.

**ВІДПОВІСТИ**


 Множина – сукупність об'єктів, які відрізняються за деякою ознакою

Рисунок 4.6 – Сторінка із запитаннями

В темплейті в директиві `v-if` перевіряється умова, за якої буде виведено той чи інший тип запитання:

```
v-if="currentQuestionType === 'single'"
```

`currentQuestionType` - тип поточного запитання, який може бути “single”, “multiple” або “dnd”. Це обчислювана властивість - `this.quiz.questions[this.currentQuestion]?.type`. `quiz.questions` - “реактивний” масив, який був зкопійований з `quiz.json`. `currentQuestion` - індекс масиву.

При виборі відповіді, `value` обраного `radio button` записується в змінну `currentAnswerValue`. При натисканні кнопки “ВІДПОВІСТИ” - викликається обробник події `@click="next"`

```

if (this.isCurrentAnswerCorrect) {
  this.goNext();
} else if (!this.isCurrentAnswerCorrect && this.currentAnswerValue) {
  this.showCorrectAnswer();
}

```

в якому перевіряється чи обраний варіант правильний вірний. Якщо правильний - викликається метод `goNext()` (перехід до наступного запитання):

```

goNext() {
  if (this.lastQuestion) this.$router.push('/finish');
  this.currentQuestion = this.currentQuestion + 1;
},

```

якщо це останнє питання - перехід до останньої сторінки - `"/finish"`, інакше - індекс масиву `quiz.questsins` інкрементується.

Якщо варіант обрано неправильно, показується `notification` з правильною відповіддю.

```

showCorrectAnswer() {
  this.$q.notify( opts: {
    type: 'negative',
    message: this.quiz.questions[this.currentQuestion].correctAnswer,
    html: true,
  });
},

```

Якщо відповідь не була обрана, `notification` не буде показано.

3. На останній сторінці користувач може бачити повідомлення, що тренування завершено (рис. 4.7).

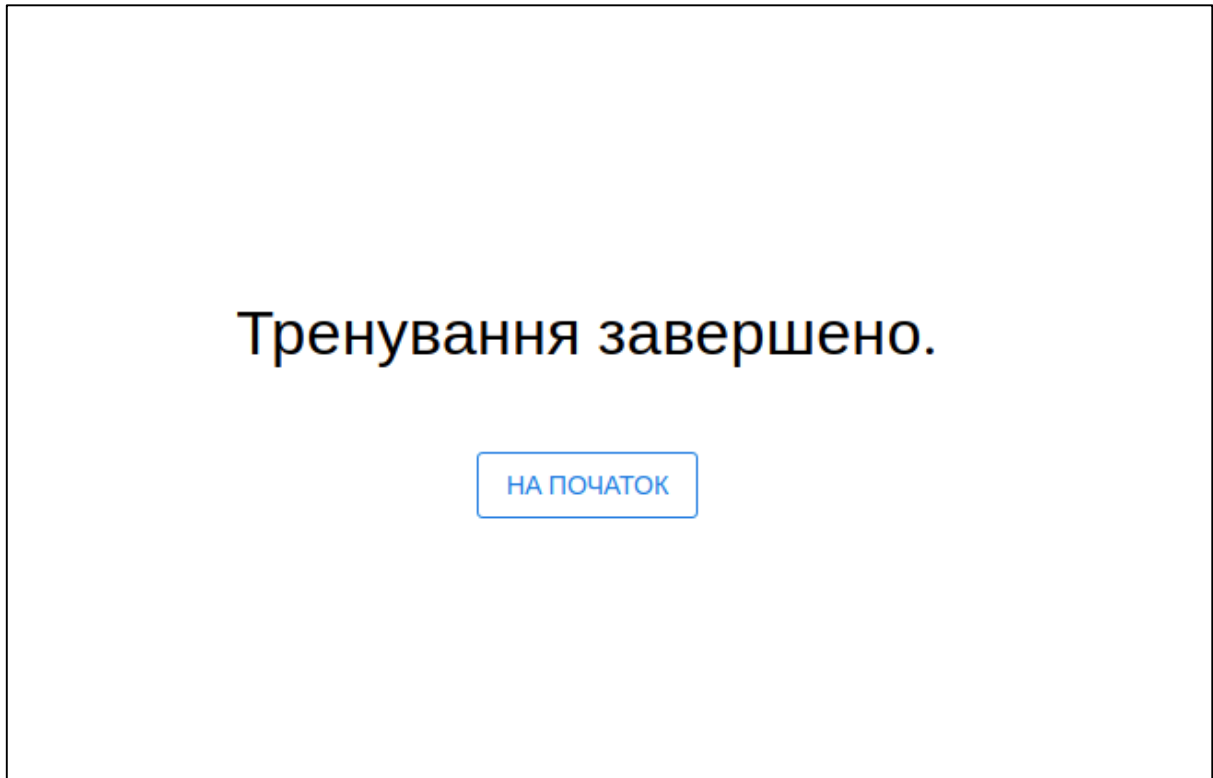


Рисунок 4.7 – Остання сторінка

При натисканні на кнопку “НА ПОЧАТОК” - відбувається перехід на початкову сторінку.

### **4.3. Опис роботи тренажера**

Розглянемо роботу програми на прикладі. Виводиться питання, потрібно обрати відповідь із запропонованих варіантів (рис. 4.8).

Виводиться наступне питання (рис. 4.9), знову вибираємо відповідь. Якщо вона вірна, то перехід до наступного.

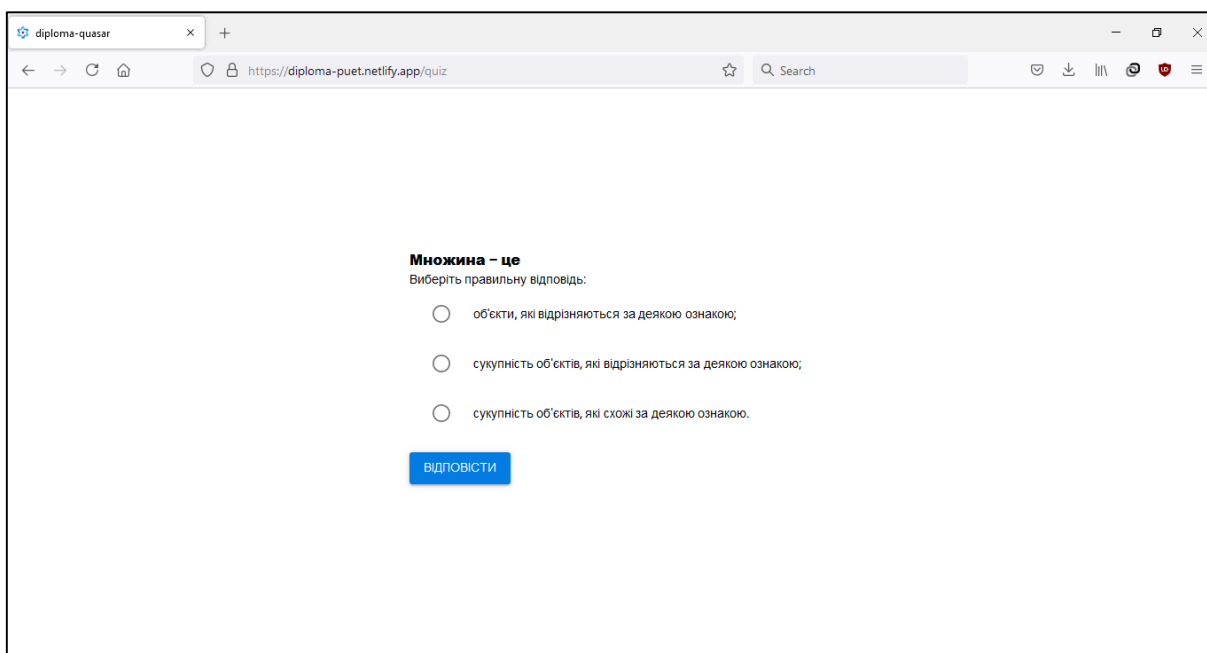


Рисунок 4.8 – Перше питання

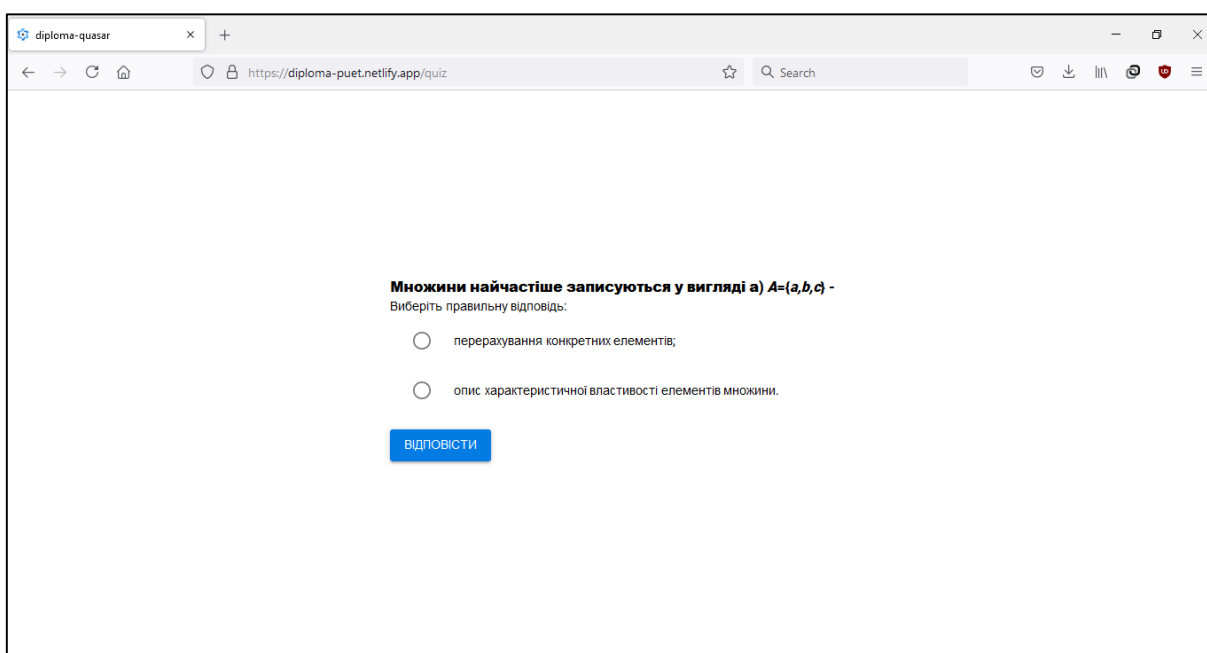


Рисунок 4.9 – Друге питання

На деяких кроках надається можливість вибрати декілька варіантів (рис. 4.10).

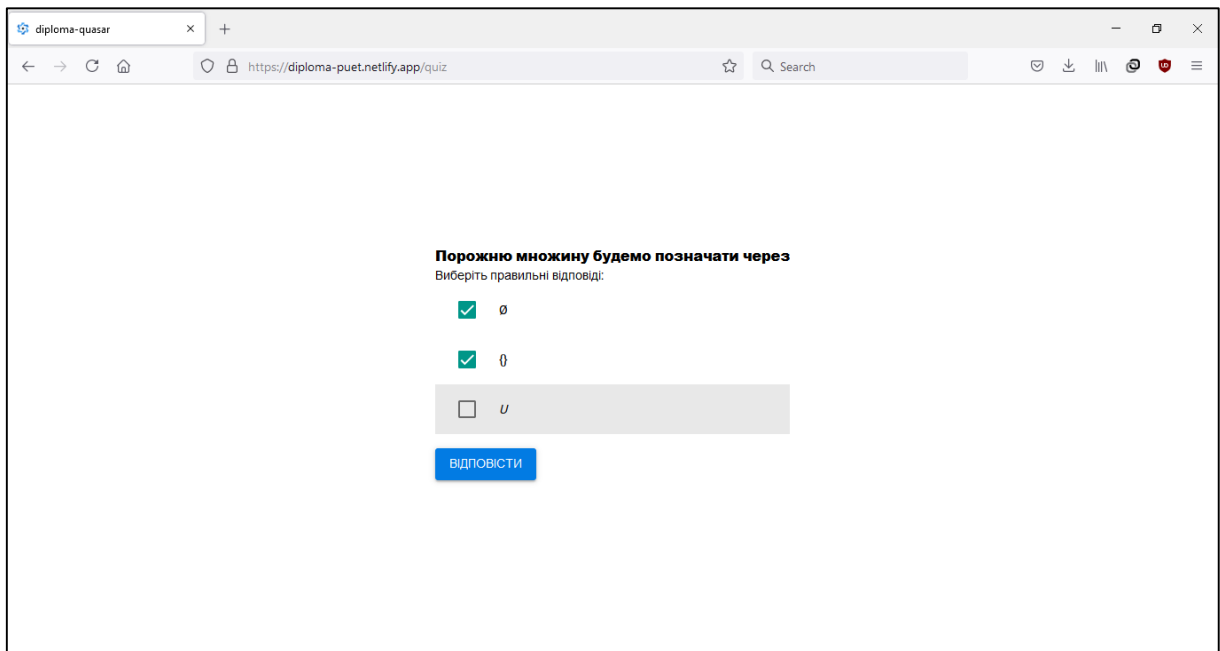


Рисунок 4.10 – Четверте питання

Продовжуючи далі, будуть питання на встановлення відповідності. Для цього необхідно перетягти варіанти на потрібні місця (рис. 4.11).

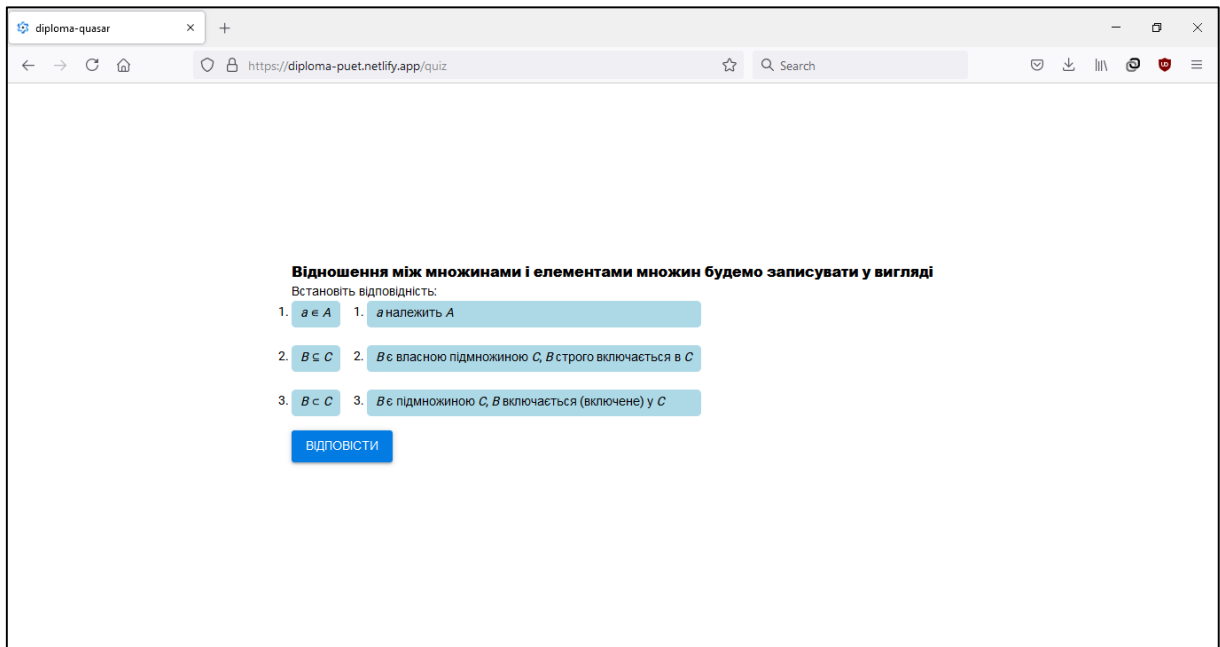


Рисунок 4.11 – Встановлення відповідності

На останній сторінці виводиться повідомлення, що тренування завершено (рис. 4.7).

## ВИСНОВКИ

Впровадження в освіту технологій дистанційного навчання значно підвищує продуктивні можливості вищої школи, розширює інформаційне освітнє середовище, збільшує можливості комунікації студентів і педагогів з колегами інших ВНЗ, надає доступ до світових інформаційних ресурсів. Під впливом дистанційної освіти, яка не в змозі повністю перебрати на себе класичну місію традиційного університету, нині формується модель гібридного університету, яка покликана поєднати аудиторне та онлайн навчання. Глибина розуміння інформації повинна стати індикатором ступеня її опрацювання й засвоєння, а це неможливо без комплексного поєднання традиційних видів навчання з дистанційними технологіями.

Розглянемо основні завдання роботи:

- проведено аналіз переваг використання тренажерів;
- проведено вибір програмних засобів для реалізації програми;
- розглянуто теоретичні відомості за обраною темою;
- розроблено алгоритм тренажеру з теми «Математичні основи дисципліни»;
- розроблено тренажер;
- описано процес розробки тренажеру;
- описано роботу програми на прикладі.

Тренажер складається з трьох сторінок:

1. Початкова сторінка, на якій можна переглянути або завантажити матеріал дипломної роботи та розпочати тренування.
2. Сторінка із запитаннями .
3. Сторінка з повідомленням, що тренування завершено.

Всі запитання знаходяться в простому JSON файлі та з легкістю можуть бути відредаговані чи доповнені.

Тренажер розрахований на запитання трьох типів:

- з однією правильною відповіддю - відображається radio group, в якій потрібно обрати правильну відповідь;
- з декількома правильними відповідями - відображаються checkboxes, з яких потрібно обрати правильні;
- встановлення відповідності - відображаються 2 вертикальні списки, які можна перетягувати за допомогою drag-n-drop.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації до виконання бакалаврської роботи для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки та інформаційні технології» галузь знань - 12 «Інформаційні технології» / О.О.(Олег) Ємець. – Полтава : РВВ ПУЕТ, 2011. – 71 с.
2. Каленіченко Л.І. Заочне та дистанційне навчання: порівняльний аналіз // Вісник Харківського національного університету внутрішніх справ. / Л.І. Каленіченко– 2009. – 28 с.
3. ATutor User Documentation. [Електронний ресурс]. – Режим доступу: <http://help.atutor.ca/general/>.
4. Система управління обучением Claroline\_\_Журнал СА 7.2008 [Електронний ресурс] – Режим доступа: <http://samag.ru/archive/article/1733>
5. Інструкція студенту / Київський національний економічний університет імені Вадима Гетьмана. [Електронний ресурс]. – Режим доступу: <https://docviewer.yandex.ua/view/43016211>
6. Черненко О.О. Електронний навчально-методичний посібник для самостійного вивчення навчальної дисципліни «Теорія програмування» для студентів напряму 6.040302 «Інформатика» – Режим доступу: <http://tprogr.ho.ua>.
7. Бабій М.С. Теорія програмування: Навчальний посібник [Електронний ресурс] / М.С. Бабій, О.П. Чекалов.– Суми: Вид-во СумДУ, 2009. – 181 с.
8. Нікітченко М.С. Теоретичні основи програмування: Навчальний посібник [Електронний ресурс] / М.С. Нікітченко. – Київ: КНУ ім. Т.Г. Шевченка, 2009. – 200 с. – Режим доступу: <http://tpp.unicyb.kiev.ua/doc/TOP.pdf>.



9. Нікітченко М.С. Теоретичні основи програмування: Навчальний посібник [Електронний ресурс] / М.С. Нікітченко. – Київ: КНУ ім. Т.Г. Шевченка, 2009. – 200 с. – Режим доступу: <http://ttp.unicyb.kiev.ua/doc/TOP.pdf>.
10. Netlify: Develop & deploy the best web experiences in record ... [Електронний ресурс] . – Режим доступу: <https://www.netlify.com/>
11. Vue Router - Официальный маршрутизатор для Vue.js. [Електронний ресурс] . – Режим доступу: <https://router.vuejs.org/ru/>
12. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.

## ДОДАТОК А